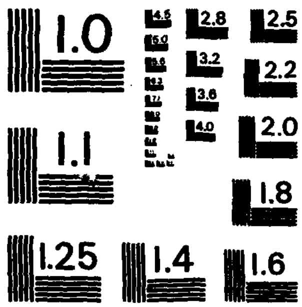
AD-A126 562 PROTOCOL LABORATORY AND TEST FACILITY IMPLEMENTATION PLAN(U) SYSTEM DEVELOPMENT CORP SANTA MONICA CA C M SWITZKY 08 JAN 82 SDC-TM-7038/218/00 1/1

UNCLASSIFIED DCA100-80-C-0044 F/G 5/1 NL

END
DATE
FILMED


1.0
4.5
5.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

SDC

System Development Corporation

2500 Colorado Avenue, Santa Monica, CA 90406, Telephone (213) 820-4111

series base no./vol./reissue

TM- 7038/218/00

author Carl M. Switzky

technical David J. Kaufman

release David J. Kaufman

for Charles A. Savant

date 1/8/82

TM a working paper

This document was produced by System Development Corporation in performance of Contract DCA100-80-C-0044

DCEC PROTOCOLS STANDARDIZATION PROGRAM

PROTOCOL LABORATORY AND TEST FACILITY IMPLEMENTATION PLAN

ABSTRACT

This paper presents the implementation plan for the DCEC Protocol Laboratory. Issues addressed include the staffing and location of the laboratory, the development of the laboratory in phases, the procurement of hardware and software, and the system integration.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 7038/218/00	2. GOVT ACCESSION NO. AD-A112562	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Protocol Laboratory and Test Facility Implementation Plan		5. TYPE OF REPORT & PERIOD COVERED interim technical report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carl M. Switzky		8. CONTRACT OR GRANT NUMBER(s) DCA100-80-C-0044
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation 2500 Colorado Ave. Santa Monica, CA 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P.E. 33126K Task 105A.558
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Engineering Center Switched Network Engineering Directorate 1860 Wiehle Ave., Reston, VA 22090		12. REPORT DATE 8 Jan 82
		13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) N/A		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

This document represents results of interim studies which are continuing at the DCEC of DCA.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Protocols, Data Communications, Data Networks, Protocol Standardization, Protocol Laboratory & Testing.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents the implementation plan for the DCEC Protocol Laboratory. Issues addressed include the staffing and location of the laboratory in phases, the procurement of hardware and software, and the system integration.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

1. INTRODUCTION

The purpose of the protocol laboratory is to assist in the production of DoD standard protocol specifications and allow the certification of implementations of DoD standard protocols done outside the laboratory. The objective of developing DoD standard protocols is to produce specifications that define the desired services, are free of errors, and can be efficiently implemented. The objective of certification is to verify the conformance of a protocol implementation to its specification.

The laboratory requirements document [1] defines the elements that make up the laboratory and the procedures followed within the laboratory to meet its objectives. The implementation plan discusses the laboratory development. Issues addressed include staffing, location, phasing, procurement of hardware and software, and integration of the components. For the purposes of this report, familiarity with the laboratory requirements document is assumed.

2. LABORATORY STAFFING

In order to provide some of the context for the Protocol Laboratory implementation plan, it is important to identify the categories of personnel that will be involved in its operation and the roles that these individuals will play. The effectiveness of the protocol laboratory will be largely dependent on the quality of these personnel and their understanding of the nature of the laboratory operation.

Laboratory roles may be divided into three general categories: laboratory users, system support personnel, and operations personnel.

The most important of these categories is the laboratory user. This category includes people that make use of the laboratory to aid in the specification of protocol standards and to test the conformance of an implementation to the standard. The laboratory users utilize the available tools and facilities as a protocol is developed and proceeds through its life cycle. This utilization starts when the user begins the specification of the services that a protocol provides and requires, follows the development through the specification and examination of alternative protocol mechanisms, allows the simulation of proposed protocols under realistic testing conditions, and provides the capability of testing non-laboratory implementations for verification of their conformance to the standard.

Support of the laboratory users requires tools and procedures for specification and testing of protocols. The design, implementation, and maintnance of these tools is the responsibility of the second category of laboratory staff -- the system support personnel. These individuals must work closely with the laboratory users to insure that the necessary tool are available and are convenient to use. Because not all the tools for the protocol laboratory will be immediately available, the system support personnel must plan the development of the tools with a schedule that will provide maximum benefit to the users as well as minimum interruption of ongoing efforts.

A third category of personnel is necessary to carry on the day-to-day operations of the laboratory -- the operations personnel. These persons are responsible for the routine maintenance and support of the laboratory hardware, maintenance of any general-purpose operating systems, as well as archiving and backup operations.

The decision as to whether to use DCEC or contractor personnel for staffing of the laboratory will be based on various criteria including personnel availability, laboratory location, and the category of personnel being considered.

In general, it is expected that laboratory users would be employees of contractors that have been tasked with the specification of DoD protocols. These personnel would make use of the laboratory facilities either at the laboratory site, or if such capability were supported, from remote sites. The certification of the contractor-produced implementations of standard protocols could be done either by other contractors, or by government representatives.

The system support work that is necessary for the laboratory will best be performed by the contractor tasked with the laboratory implementation. This contractor needs to be familiar with the laboratory requirements in order to produce a realistic and responsive development schedule. Because of the nature of the tools necessary for the laboratory, development of individual tools may take place away from the laboratory site, with the installation occurring only after the tools have undergone extensive testing and evaluation without disrupting the operational facilities. It is therefore not necessary for the contractor to locate the entire development team at the laboratory site.

Day-to-day operations of the Protocol Laboratory could be performed either by contractor or government supplied personnel. The tasks to be performed are specified by the system support team in conjunction with the requirements of the laboratory users. These personnel must be located at the laboratory site.

3. LABORATORY LOCATION

The laboratory must be situated at a location that is capable of supporting the necessary equipment and is also convenient for its users. Possible locations for the protocol laboratory include the DCEC facility at Reston, Virginia, a separate facility in the Washington area leased for the laboratory, or a contractor's site. Alternately, the modular nature of the laboratory design also allows for a distributed laboratory to be implemented at multiple sites. Each site would provide all or some of the laboratory capabilities, and coordination between the sites would be provided by network gateways.

The selection of the laboratory location necessitates evaluation of several factors, including the convenience of the facility's location for its users and staff, the relative costs of the alternative locations, the amount of space available to support the necessary equipment, the ability of the sites to meet any special security requirements, and proximity to other network equipment.

A single laboratory at the DCEC facility in Reston allows close monitoring and control of laboratory usage by DCEC personnel and is readily accessible to other government agencies. Since many potential DoD contractors maintain Washington area offices, the Reston location is convenient to these users. In addition, the Exploratory Data Network (EDN), an adjunct facility to the protocol laboratory, is situated within the DCEC facility. The laboratory could be colocated with the EDN, although additional space would need to be allocated. The DCEC location can provide the appropriate security environment for special equipment that may be incorporated into the laboratory. Unfortunately, the Reston facility may not be able to meet the space requirements for the laboratory. Naturally, if physical space requirements could not be met by the Reston facility, a separate facility in the area could be leased. This option would provide many of the same advantages of the DCEC Reston location, but may render close monitoring by government personnel more difficult. In addition, the cost of leasing the necessary space may be higher than utilizing available space at DCEC, and leased space would also require upgrades to meet special security requirements necessary for the development of security related protocols.

The laboratory design provides for remote accessibility of the laboratory facilities. This provision mitigates the requirement that laboratory users be at the installation site. Furthermore, portions of the laboratory could be located at the facilities of the contractor responsible for laboratory development and support. In this way, ready access to the laboratory is provided to all system support personnel.

The diverse nature of the laboratory makes it likely that multiple contractors will utilize the laboratory facilities. Typically, a single contractor should be responsible for the system support role of designing, implementing, and maintaining laboratory tools. Other contractors may be responsible independently for the development of individual protocol standards or for implementation of these standards in different environments. Additional contractors may be responsible for the certification of these implementations. These different laboratory roles need not all be performed at the same physical location. The local network design of the laboratory supports a distributed laboratory in which multiple development and test machines may exist at different interconnected sites.

4. PHASING OF IMPLEMENTATION

Because of the complexity and sophistication of the required laboratory, it is both necessary and desirable to implement the facility on a phased basis. Incremental development and growth of a hardware/software system provides for orderly integration of components and for change in emphasis as experience is gained in using the developed system. In addition, a phased laboratory development can be engineered so that at the finish of each phase, a complete and integrated system is available for use.

The phasing of the laboratory must also be geared to the development of new standard protocols. Although the evaluation of all protocols and their specifications will benefit from the analysis capability of the complete

laboratory, the facilities implemented in the early phase should be those that will have maximum applicability to protocols currently under development. For example, automated test generation, a valuable tool for performing reachability analysis, is not as necessary in the early phase as the capability for accurately and conveniently specifying protocols such as TCP and IP. However, when new, untested protocols are being developed, this tool will be essential to properly test proposed versions of the protocols.

When viewed in conjunction with the rest of the DCEC Protocols Standardization Program and the schedule of protocol development, the laboratory logically divides into three phases. The Initial Phase includes the acquisition of the hardware for the initial laboratory configuration and the integration of available software to perform basic laboratory functions. The Intermediate Phase enhances the capabilities provided in the initial hardware/software system by incorporating custom development, analysis, and testing tools. The Advanced Phase furthers these capabilities with the inclusion of more sophisticated tools and allows for additional laboratory growth through geographic distribution.

4.1 Initial Phase

The goals of the Initial Phase of the protocol laboratory are to demonstrate its initial operating capability and to produce an extensible hardware/software base on which further phases of the laboratory can be built. These goals will be achieved through the integration of available hardware and software into development and testing systems, tied together by a multichannel local network. The emphasis during the Initial Phase will be on using available components whenever possible so as to minimize efforts and risks associated with the development of new software and hardware.

4.1.1 Protocol Development System

The protocol development system for the Initial Phase will consist of a general purpose computer supporting available editors, formatters, compilers, linkers, utilities, a document control system, and a message system. A syntax-directed editor and service simulator to aid in the development of state machine specifications are also included.

The editors and formatters assist in the preparation of protocol service and mechanism specifications by allowing the user to combine together the desired text and state machine descriptions. A syntax-directed editor enforces syntax rules and insures consistency of terminology throughout the document. The syntax for protocol specification is given in [2]. The service simulator helps the designer to verify that interfaces behave as they are intended.

The compilers and linkers allow users to prepare implementations of protocols for evaluation in the test environment. The tools, augmented with standardised subroutines, allow the instrumentation of protocols to help in the analysis of their behavior and performance. The linkers are used to combine upper and lower level routines necessary to form the test configuration.

The utilities and document control system are necessary to keep track of the various versions of specifications and implementations that are produced in the laboratory. These tools assure that the current versions are available to laboratory users, and that older versions may also be obtained if desired. The document control system must also be applied to each of the laboratory software components and related documentation.

The message system facilitates communication between laboratory users, the system support group, and other persons involved in the laboratory. It can be used for scheduling the use of the facilities, distributing results of testing, and for communicating with persons outside the laboratory.

For the Initial Phase, much of the required software and a suitable operating system is commercially available. Unix provides a general purpose operating system and much of the necessary software. It also provides a convenient development environment both for protocol specifications and laboratory components. Other useful Unix-compatible software packages are available from a wide range of sources.

Initial Phase software development for the Protocol Development System includes the syntax-directed editor and the service simulator. The syntax-directed editor will be adapted to the formats used in the service and mechanism specifications. These formats are specified in [2]. The service simulator will accept state machine specifications, a probability assignment for non-deterministic events, and a test script of service events. These tools are described in more detail in Sections 4.2 and 4.3 of [1]. Development effort will also be required to support protocol instrumentation primitives in compilers for the test environment. Finally, an interface to the local network will have to be added to the Unix system.

4.1.2 Test Environment

The test environment supports the evaluation of protocols by exercising their implementations between different test units across a controlled network environment. Each test unit is composed of a target environment processor, an adaptor environment processor, and a pair of channel interfaces to the local network.

The target environment processor supports a protocol implementation and other test control software. The laboratory design allows different types of representative processors to be employed, allowing for the preparation and testing of implementations on processors typical of those that will have to support the protocol. This capability enables the designer to determine the difficulty of implementing a proposed mechanism for a specific type of processor. To simplify efforts during the Initial Phase, only one type of processor will be used.

The adaptor environment processor allows the emulation of the network characteristics necessary for protocol testing. Characteristics controlled in the Initial Phase include the data rate, the delay, and the reliability of the underlying network. By controlling these characteristics, the testing can take into account the ability of the protocol under test to handle different

data rates, delays inherent in internetting, and problems that occur from the corruption of data over links.

The channel interfaces provide communication paths between the test units that are exercising protocols and provide the control of the tests by the development system. Separate channels keep the access to the local network for sample network simulation separate from access to the local network for test control. The test control path will be used for starting and stopping tests, controlling the network characteristics, and for obtaining results from tests.

The major development effort for the Initial Phase will take place in the Test Environment. The first part of this development effort is the assembly of the test units. This requires selection of the processors for the target and adaptor environments, the processors that will support the network interfaces, the bus structure that will be used for internal communication, and the power supply and packaging of the components. The next part of the development effort is the design and implementation of operating systems suitable for supporting implementations under test, and the processes in the adaptor environment. The characteristics of these operating systems are similar because of the requirement for approved protocols to become part of the adaptor environment by installation within that processor. More detailed specification of the test unit operating system requirements can be found in Section 6.3.1 of [1]. Other system level software to be developed for the Initial Phase include test monitoring and control programs and loaders for porting modules into the test environment.

4.1.3 Local Network

The local network for the protocol laboratory will provide the physical connections between the development system and the test units. A multi-channel system provides independent channels for both network simulation and test control. The extra channels provide for additional growth in later phases.

Many local network offerings exist on the market today. The requirements for the protocol laboratory can be met by a wide-bandwidth broadband coaxial cable local network. This type of network can dedicate multiple frequency-derived channels which can be used both for inter-processor communication and for simulation of a wide variety of communication networks. For the Initial Phase, the installation of cable and other associated local network plant will be performed.

4.2 Intermediate Phase

The goals of the Intermediate Phase are to enhance the capabilities of the laboratory for the testing and evaluation of new protocols and to permit field testing of external protocol implementations. These goals will be achieved by the addition of specially designed software tools to the laboratory. The hardware configuration developed during the Initial Phase will remain basically unchanged except for the addition of other test units with different target environment processors.

4.2.1 Intermediate Software Tools

Software tools that will be developed during the Intermediate Phase include template-based test generators, interactive state machine interpreters, peer interpreters, automated laboratory implementation generators, and analysis tools. In the test units, TCP and IP protocols will be migrated into the adaptor environments so that other protocols can be tested in the isolated environment of the target processor.

Use of the service simulator for the Initial Phase requires the specification of the service interface, an assignment of probabilities for non-deterministic events, and a manually generated sequence of service events to drive the simulator. This sequence of events will also be used as a basis for testing protocol mechanisms and implementations. It is desirable during the Intermediate Phase to use templates to assist in the generation of test event sequences. Templates identify the general pattern of testing to be performed and provide some automation of the test event sequences. They are described in greater detail in Section 4.2 of [1].

The next step in enhanced protocol testing is the ability to more carefully evaluate a proposed protocol mechanism. This evaluation is achieved by using an interactive state machine interpreter that allows exercising a state machine with control over the interactions at the upper, lower, and system interfaces. Test scenarios are based on those used for testing the service interfaces, and service simulators are used for the lower level interface. The interpreter also allows the execution of test sequences conceived on the fly by the tester. This testing is described in more detail in Section 4.3 of [1].

To further test a proposed mechanism requires that a peer interpreter be used to eliminate the need for explicit control over the lower level interface. The peer interpreter will allow the symbolic execution of the proposed protocol before effort is expended on an implementation.

In order to more fully test a proposed mechanism it is necessary to produce an instrumented implementation of the protocol for the test environment. Automated laboratory implementation generators will simplify this effort for the tester by using the formal state machine description of the mechanism as input, and generating code for the state machine portion of the implementation. The generated code is not intended to be efficient, but instead to provide the tester with a quick implementation of the protocol so that its basic operating characteristics can be assessed.

Finally, these additional tools will be greatly enhanced by automated analysis tools that record results of the tests and provide summaries of state counts, message counts, and behavior of the system interfaces.

4.2.2 Test Environment Modifications

Because the Intermediate Phase is intended to coincide with the development of protocols that will be using TCP and IP as a base, the test environment must be modified so that implementations of these protocols are migrated into the

adaptor processor environment. The implementations of these protocols then become part of the controlled environment that is necessary for the higher level protocols. Elements of these protocols that need to be controlled include their throughput capability, delay in opening connections, and segment size.

Testing of protocols that involve multiple connections between more than two sites may necessitate the addition of more test units to the test environment. An example of this requirement is an FTP three-party transfer involving control, donar, and recipient sites. During this phase it is also appropriate to include additional types of target environment processors.

4.2.3 Field Testing of External Implementations

It is necessary in the Intermediate Phase to incorporate the capability of testing and certifying externally produced implementations of protocols stand ardized in the laboratory. This type of testing is aimed at certifying tha two external implementations of a protocol will communicate with each oth over actual lower level protocol implementations and communications facili ties.

For the Intermediate Phase, the field testing approach is to test an impleme tation at the site through the use of a test driver and available lower leve protocols. The test driver controls the upper layer interface of the protocol under test, and is built to laboratory specifications. It is controlled by the laboratory through an independent connection and facilitates testing similar to that done with protocol peer entities.

4.3 Advanced Phase

The goals for the Advanced Phase include support for more in-depth analysis of proposed protocols and geographic distribution of the lab capabilities. Geographic distribution of the Protocol Laboratory has two functions: to enhance field testing of external implementations of standardized protocols, and to coordinate different protocol laboratory locations.

4.3.1 Protocol Analysis

More in-depth analysis of a proposed protocol is accomplished through the development of automated test generation techniques. These techniques are a compromise between impractical exhaustive-search testing, and non-comprehensive, tester driven walk-throughs. The interactive state machine simulator developed for the Intermediate Phase allows the protocol designer to drive the protocol through a given sequence of desirable transitions, but it does not help in verifying the reachability of all protocol states. In order to perform reachability analysis of a protocol specification, it is necessary to determine whether a state can be reached following a specific sequence of transitions, in response to a specific sequence of events. Automatic generation of event sequences can be done by symbolic manipulation of the enabling predicates associated with desired transitions. The Advanced Phase incorporates tools to generate event sequences that will assist in exposing flaws in protocol specifications and assuring conformance exists between mechanism

specifications and their implementations. The automatic generation of event sequences is described in more detail in Section 5.1 of [1].

The Advanced Phase also provides the capabilities of testing for deadlock conditions and verifying that mechanism specifications correspond to service specifications. Deadlock detection involves peer interaction analysis, applying automated test event generation techniques to the symbolic protocol execution environment.

4.3.2 Geographic Distribution

The testing of external implementations during the Intermediate Phase is limited in its capability to fully certify an implementation because of inherent limitations of using lower level protocols and networks that are not completely under laboratory control. For the Advanced Phase, the external testing is enhanced by the development of additional software that monitors the behavior at the interface with lower level protocols. The results of such passive monitoring are communicated to the laboratory following a test run and are incorporated in the analysis of the protocol. Also available for the Advanced Phase is software that replaces the lower level protocol and provides control over test events at both upper and lower interfaces.

The Advanced Phase also permits geographic distribution of laboratory components through the establishment of other sites as extensions of the Protocol Laboratory. To support this activity requires the development of procedures that coordinate and control the activities occurring at each of the sites. Multiple sites also require network connectivity so that documents, programs, and standards can be exchanged and kept up to date.

5. SELECTION AND PROCUREMENT OF HARDWARE

A critical section of the laboratory implementation is the selection of hardware that will meet the processing and communications requirements, while offering reliability, flexibility, and availability. The laboratory is composed of three types of hardware elements, each of which has its own characteristics: the development system, the test environment, and the local network.

The development system can be acquired as an off-the-shelf system. It consists of a main processor, associated online and offline storage, terminals, printers, and the capability for remote access by communications networks and other local networks. Because the development system is necessary from the inception of the laboratory, procurement of it should begin as soon as possible.

Unlike the development system, the test environment is not available as a complete package. As mentioned earlier, the test environment is composed of a number of test units each of which will have to be fabricated. The criteria for selecting the components of the test units are familiarity and experience of the system support personnel, capabilities of the bus structure, and availability of necessary card level products. Processors for the target and

adaptor environments should be the same to simplify the migration of protocols from one board to the other. Currently, an Intel Multibus based system seems desirable because of the supportable data rates and addressing structures, and the availability of a large number of compatible off-the shelf products. Procurement of pieces for the test units should coincide with the capability for their assembly and testing. This production of the test units also requires other hardware equipment for testing interfaces, line levels, power supplies, and other related electrical properties.

The local network needs to be broadband based in order to separate development and testing channels. The bandwidth needs to be high enough so that the data rate characteristics of a variety of communications networks can be simulated. Finally, the local net interfaces must be available to all the connected processors. Installation of the local network must be performed by the time the test units are functional.

6. SELECTION AND PROCUREMENT OF SOFTWARE

Software selection for the Protocol Laboratory is based on the analysis of requirements of each necessary item. Whenever possible, commercially available software will be used. Criteria for the selection of software include applicability of the software to the specific problem, previous experience of the system support personnel with the software, extensibility of the software for the laboratory specific functions, and support of the software by the supplier.

Whenever laboratory software requirements cannot be met by commercially available software, the items will be custom developed for the laboratory. This development must be performed in a carefully controlled manner. After the specific functions for a software item are identified, a detailed specification of the software is produced and reviewed. Laboratory support personnel are responsible for the design, programming, testing, and installation of new software modules. In addition, program and user documentation requirements must be met.

In order to keep track of different versions of software items, and to facilitate maintenance of multiple protocol laboratory sites, the documentation control system of the laboratory must be employed. Every time a new version of a commercially supplied item is obtained or an updated version of a locally developed item is produced, it must be recorded in the document control system, along with other pertinent information such as the responsible party, date, and reasons. Although this type of documentation control is limited to software, it permits reconstruction of previous tests should the current system no longer support them.

7. SYSTEM INTEGRATION

System integration is performed in an incremental fashion, so that at each step, all preceding increments will have been tested, and new problems can be isolated.

System integration for the development system begins with installation and check-out of the selected development system hardware. Installation of the operating system follows, with partitioning of the available storage space into a file system structure that allows each of the laboratory users private work spaces as well as access to common laboratory facilities. After the operating system and the associated software tools are installed, the laboratory is available for operation, and in parallel, the development of custom laboratory software begins. Before new software is ready for installation, it must be tested by selected laboratory users in an isolated environment so that any serious problems can be identified before general release.

For the test environment, system integration must also proceed in a careful, controlled manner. Each of the components that make up the test units must be individually tested before they are assembled in order to prevent damage to other components. Once a complete test unit has been assembled, a set of diagnostics can be developed that verifies proper operation of the hardware, and localizes problems when they occur. Operating system development is followed by the installation of the software for loading and controlling the test units. The next step in the integration is the verification that cross compilers for the test units produce valid programs, and the demonstration that loading and control procedures properly function.

An essential part of the system integration is the documentation of all components of the system and the notification to users of any system changes. Because of the close coordination that exists both between the users of the laboratory and the system supporters, and between the development system and the test environment, it is critical that system integration be performed in a manner that minimizes disruption of the system and its users.

8. REFERENCES

[1] Sytek Inc., "Requirements for the Protocol Laboratory and Test Facility," DCEC Protocols Standardization Program, System Development Corporation TM-7038/214/00, December 1981.

[2] G. A. Simon, "Protocol Specification Report," DCEC Protocols Standardization Program, System Development Corporation TM-7038/204/00, July 1981.

ATE

LME

8